

# PATENT ABSTRACTS OF JAPAN

(11) Publication number : 05-241934

(43) Date of publication of application : 21.09.1993

(51) Int. CI. G06F 12/00

G06F 15/16

(21) Application number : 04-042508

(71) Applicant : TOSHIBA CORP

(22) Date of filing : 28.02.1992

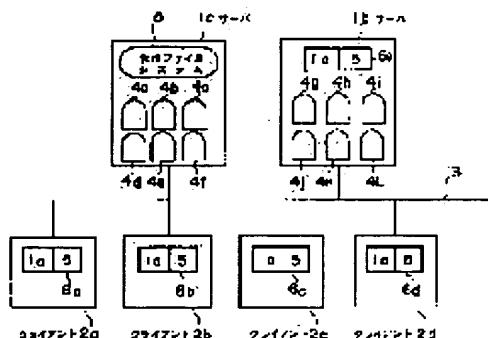
(72) Inventor : SEKIDO KAZUNORI

## (54) COMPUTE SYSTEM

### (57) Abstract:

**PURPOSE:** To handle all the files as the same file by a client regardless of the state of really arranging files to be managed in tree structure.

**CONSTITUTION:** A virtual file system 5 provided at a server 1a manages files in the entire computer system by one tree structure, and respective clients 2a-2d issue requests only to the virtual file system 5 regardless of the state of really arranging the files even when any file is used. All the file access requests from the clients 2a-2d are received by the virtual file system 5 and the file system of the server, where that file exists, is investigated according to the tree structure. Therefore, it is not necessary for the respective clients 2a-2d to manage a mount table, and flexible file management with a system change caused by the addition of servers or the move of files can be easily realized.



## LEGAL STATUS

[Date of request for examination] 16.09.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted]

[registration]  
[Date of final disposal for application]  
[Patent number] 3300399  
[Date of registration] 19.04.2002  
[Number of appeal against examiner's  
decision of rejection]  
[Date of requesting appeal against  
examiner's decision of rejection]  
[Date of extinction of right]

Copyright (C) ; 1998, 2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-241934

(43)公開日 平成5年(1993)9月21日

(51)Int.Cl.<sup>6</sup>

G 06 F 12/00  
15/16

識別記号 庁内整理番号  
5 4 5 B 7232-5B  
3 7 0 M 9190-5L

F I

技術表示箇所

(21)出願番号

特願平4-42508

(22)出願日

平成4年(1992)2月28日

審査請求 未請求 請求項の数3(全14頁)

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 関戸 一紀

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝総合研究所内

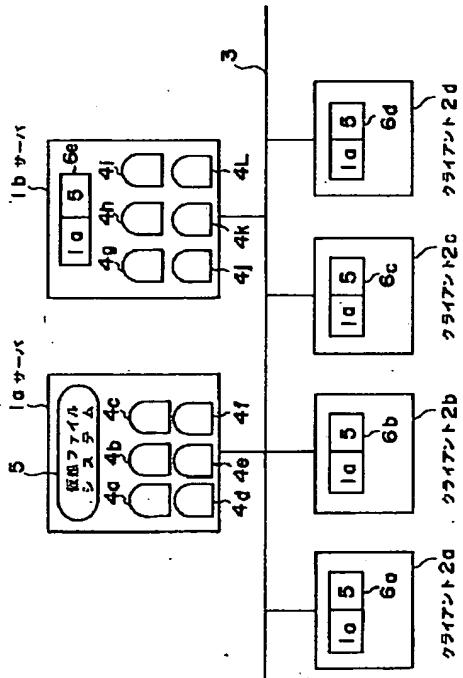
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 計算機システム

(57)【要約】

【目的】木構造で管理されるファイルの実際の配置状態に関係なくクライアントが全てのファイルを同一ファイルとして扱えるようにする。

【構成】サーバ1aに設けられた仮想ファイルシステム5は、計算機システム全体のファイルを1つの木構造で管理しており、各クライアント2a～2dは、どのファイルを使用する場合でも、ファイルの実際の配置状態に関係なく、仮想ファイルシステム5に対してのみ要求を発行する。クライアント2a～2dからのファイルアクセス要求はすべて仮想ファイルシステム5で受け付けられ、木構造に従ってそのファイルが存在するサーバのファイルシステムが調べられる。このため、各クライアント2a～2dは、マウント表の管理を行う必要がなくなり、サーバの追加等によるシステム変更やファイルの移動を伴う柔軟なファイル管理を容易に実現できる。



## 【特許請求の範囲】

【請求項1】 1以上のサーバ計算機とこのサーバ計算機が提供するファイルサービスを共有する1以上のクライアント計算機から構成されたサーバ/クライアント型の計算機システムにおいて、

前記サーバ計算機の少なくとも1つは、前記各クライアントからのファイルアクセス要求をすべて受け付けるファイル管理手段を具備し、このファイル管理手段は、前記計算機システムの全てのファイルを1つの木構造で管理する手段と、前記各クライアントからアクセス要求されたファイルが存在するサーバ計算機のファイルシステムを前記木構造に従って調べ、そのサーバ計算機のファイルシステムにアクセス要求を発行する手段と、前記アクセス要求されたファイルが存在するサーバ計算機のファイルシステムから返却されるデータを前記アクセス要求元の各クライアントに送信する手段とを具備することを特徴とするサーバ/クライアント型の計算機システム。

【請求項2】 木構造でファイル管理を行うファイルサーバを含む1以上のサーバ計算機と、前記ファイルサーバが提供するファイルサービスを共有する1以上のクライアント計算機から構成され、個々のファイルをサーバ番号、サーバ内のファイルシステム番号、そのファイルシステム内で固有のファイル識別番号から構成される識別子を用いてアクセスする計算機システムにおいて、前記木構造の接続点を記憶するマウント表と、このマウント表に対する接続点の登録・削除を行うマウント表管理手段と、前記マウント表を調べ、接続先を求めるマウント検索手段と、前記識別子から仮想的なファイル識別番号へ変換する第1の変換手段と、前記仮想的なファイル識別番号から前記識別子へ変換する第2の変換手段と、前記仮想的なファイル識別番号に対するシステムコールを対応する前記識別子に対するシステムコールへ変換するシステムコール変換手段から構成される仮想的ファイルシステムを前記サーバの少なくとも1つに具備せしめ、前記サーバおよびクライアントは各々のマウント表に前記仮想的ファイルシステムへの接続点を登録することを特徴とするサーバ/クライアント型の計算機システム。

【請求項3】 前記第2の変換手段は、前記各仮想的なファイル識別番号に対して複数の前記識別子を割り当てる手段を具備し、前記システムコール変換手段は、前記各仮想的なファイル識別番号に対するシステムコールを前記複数の識別子に対するシステムコールに変換する手段を具備することを特徴とする請求項2記載の計算機システム。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 この発明はサーバ/クライアント形式でファイルを共有する計算機システムに関する。

## 【0002】

【従来の技術】 近年、情報システムのダウンサイズ化が進んでおり、サーバ/クライアント形式のシステムが注目を集めている。サーバ/クライアントとは、高性能なマイクロプロセッサによりユーザの処理を行うクライアントと、それらで共有するデータや処理を扱うサーバから構成されるシステムの形式で、従来の汎用機と端末から構成されるシステムに比べ、高い性能や優れたインターフェース等をユーザに提供できる形式である。

【0003】 図12にファイルサーバとクライアントからなるシステムの構成を示す。13a, 13bはクライアントへファイルサービスを提供するサーバ、14a, 14b, … 14f, 14gはサーバ13a, 13bが提供するファイルサービスを共同で利用するクライアント、15はサーバ13a, 13b, クライアント14a, 14b, … 14gを接続するネットワークである。

【0004】 16a, 16b, …, 16eはサーバ13aに存在するファイルシステム、16f, 16g, …, 16jはサーバ13bに存在するファイルシステムで、それぞれ木構造を有している。17a, 17b, … 17gはそれぞれクライアント14a, 14b, … 14gに存在するマウント表、17h, 17iはサーバ13a, 13bに存在するマウント表である。これらマウント表は、ファイルアクセスのために使用されるものである。

【0005】 一般に、計算機内には多数のファイルが存在しており、名前の重複をさけるため、その管理は木構造を用いて行なわれる。この方法では、個々のファイルは葉に、ディレクトリは枝に対応する。また、個々のファイルは、ルートから途中のディレクトリ名も含めて葉ファイル名まで指定することで特定できる。これをパス名と呼ぶ。例えば、図13に示す木構造でファイルが管理されている場合を考える。

【0006】 図13において、ROOTは根、BIN, LIB, USR, ETCは第1節のディレクトリ名、ABC, DEFは第2節のディレクトリ名、XYZは第3節のディレクトリ名である。この場合、図13におけるファイル(FILE1)18は、/VSR/ABC/XYZ/FILE1のパス名で指定される。

【0007】 よって、システム内のファイルは1本の木構造である必要があるが、実際には図12に示すように、ファイルは複数のサーバに分かれ、さらに各サーバ内でも複数のファイルシステムに分かれて管理されている。そこで、これらのファイルシステムを集めて1つの木構造にするため、従来では、システム内の全クライアント14a, 14b, … 14gと全サーバ13a, 13bにマウント表をそれぞれ持たせている。

【0008】 マウント表とは、図11に示すように、1本にまとめ上げられた木構造において、それを構成する部分木構造を示す接続点と、接続された部分木構造を格

納するサーバを示すサーバ番号と、そのサーバ内のファイルシステムを示すファイルシステム番号とを対応して記憶するテーブルである。

【0009】例えば図15に示すように、／USR／ABCの点で、サーバ13aのファイルシステム16aに格納された部分木構造が接続される場合、マウント表には図14の第1エントリのように記録される。このマウント表は図12に示すように、全クライアントと全サーバにそれぞれ保持される。

【0010】次に、このシステムの動作を簡単に説明する。今、クライアント14aがバス名“／USR／ABC／XYZ／FILE1”的ファイルをオープンする場合を考える。第1に、クライアント14aは自分のマウント表17aを参照してバス名の比較を行い、その部分バス名“／USR／ABC”が一致するエントリを見つけ、サーバ13aのファイルシステム16aに目的のファイルが存在することを知る。

【0011】第2に、クライアント14aは、求ったサーバ13a、ファイルシステム16aに対して、残りのバス名“XYZ／FILE1”をオープンするリモートシステムコールを実行する。すると、サーバ13aはファイルシステム16aに対してバス名“XYZ／FILE1”的オープン処理を行い、結果として、そのファイルシステム16a内のiノード番号、例えば123をクライアント14aに返す。

【0012】このiノード番号123はサーバ13aのファイルシステム16a内の固有の番号であり、ファイルシステム16a内でのファイル識別にのみ使用されるローカルな識別番号である。クライアント14aは、図16に示すように、先に求めたサーバ番号、ファイルシステム番号と合わせて、そのiノード番号123をVノードとして管理する。

【0013】Vノード（サーバ番号、ファイルシステム番号、iノード番号）は、このクライアント／サーバ型のシステム全体から見た固有のグローバルなファイル識別番号である。

【0014】また、このVノードに対してファイルのリードを行う場合、このVノードを使って、サーバ13a、ファイルシステム16a内のiノード123のファイルリードあることが判るので、クライアント14aは、その旨をサーバ13a、ファイルシステム16aに送ればよい。

【0015】ここで、このような従来のサーバ／クライアント形式のシステムには次のような問題がある。まず、第1に、全クライアント及び全サーバがマウント表を管理しなければならないので、クライアント数が多いと、マウント表の数もそれにつれて増大される。さらに、サーバやファイルシステム数が変わった場合、全マウント表を変更しなければならず、システム管理者の作業が大変であった。

【0016】第2に、クライアントからは図16に示すようなVノード（サーバ番号、ファイルシステム番号、iノード番号）で直接ファイルをアクセスするため、ファイルを他のサーバやファイルシステムに移動したりする場合、システムを止めて行なう必要があった。

【0017】さらに、1つのファイルが複数のサーバやファイルシステムに格納できると、負荷分散やファイル配置が容易になるが、従来ではVノードで直接アクセスしていたため、アクセス先のファイルシステムが1つに固定されてしまい、ファイルの分散配置ができなかっただ。

#### 【0018】

【発明が解決しようとする課題】従来のサーバ／クライアント形式のファイルサーバでは、各クライアント、サーバが個別に木構造のマウント表を管理しているため、それらマウント表の修正に手間がかかり、サーバの追加等によるシステム変更やファイルの移動を伴う柔軟なファイル管理を行うことが困難であった。

【0019】この発明はこのような点に鑑みてなされたもので、ファイルの実際の配置状態に関係なくクライアントは1つの仮想ファイルシステムだけをアクセスする事によって任意のファイルアクセスを実現できるようにし、クライアントにおけるマウント表の管理を不要とすることによって、サーバの追加等によるシステム変更やファイルの移動を伴う柔軟なファイル管理を行うことができる計算機システムを提供することを目的とする。

#### 【0020】

【課題を解決するための手段および作用】この発明は、1以上のサーバ計算機とこのサーバ計算機が提供するファイルサービスを共有する1以上のクライアント計算機から構成されたサーバ／クライアント型の計算機システムにおいて、前記少なくとも1つのサーバ計算機に、前記各クライアントからのファイルアクセス要求をすべて受け付けるファイル管理手段を具備し、このファイル管理手段は、前記計算機システムの全てのファイルを1つの木構造で管理する手段と、前記各クライアントからアクセス要求されたファイルが存在するサーバ計算機のファイルシステムを前記木構造に従って調べ、そのサーバ計算機のファイルシステムにアクセス要求を発行する手段と、前記アクセス要求されたファイルが存在するサーバ計算機のファイルシステムから返却されるデータを前記アクセス要求元の各クライアントに送信する手段とを具備することを第1の特徴とする。

【0021】この計算機システムの構成では、ファイル管理手段が1つの木構造をすでに構成しているので、ファイルの実際の配置状態に関係なくクライアントはそのファイル管理手段だけをアクセスする事によって任意のファイルアクセスを実現できる。このため、各クライアントは全てのファイルを同一ファイルとして扱えるようになり、各クライアントにおいてマウント表の管理を行

う必要がなくなるため、サーバの追加等によるシステム変更やファイルの移動を伴う柔軟なファイル管理を実現できる。

【0022】また、この発明は、木構造でファイル管理を行うファイルサーバを含む1以上のサーバ計算機と、前記ファイルサーバが提供するファイルサービスを共有する1以上のクライアント計算機から構成され、個々のファイルをサーバ番号、サーバ内のファイルシステム番号、そのファイルシステム内で固有のファイル識別番号から構成される識別子を用いてアクセスする計算機システムにおいて、前記木構造の接続点を記憶するマウント表と、このマウント表に対する接続点の登録・削除を行うマウント表管理手段と、前記マウント表を調べ、接続先を求めるマウント検索手段と、前記識別子から仮想的なファイル識別番号へ変換する第1の変換手段と、前記仮想的なファイル識別番号から前記識別子へ変換する第2の変換手段と、前記仮想的なファイル識別番号に対するシステムコールを対応する前記識別子に対するシステムコールへ変換するシステムコール変換手段から構成される仮想的ファイルシステムを前記サーバの少なくとも1つに具備せしめ、前記サーバおよびクライアントは各々のマウント表に前記仮想的ファイルシステムへの接続点を登録することを第2の特徴とする。

【0023】この構成においては、クライアントやサーバは、サーバ番号、ファイルシステム番号、ファイル識別番号を含む識別子で直接ファイルをアクセスするのではなく、仮想的ファイル識別番号を用いて行っているので、ファイルを他のサーバやファイルシステムに移動すること等が容易に行える。また、仮想的ファイル識別番号を用いると、アクセス先のファイルシステムが1つに固定されないので、ファイルの分散配置等による負荷分散を図ることができる。

【0024】さらに、1つの仮想的ファイル識別番号に対して複数の識別子を対応させる機能を設け、システムコール変換手段に1つの仮想的ファイル識別番号に対するシステムコールを複数の識別子に対するシステムコールに変換する機能を設けることによって、複数のサーバやファイルシステムに分割して格納されたファイルを1つのファイルとしてクライアントにサービス提供できる。よって、負荷分散と、効率的ファイル配置を実現できる。

#### 【0025】

【実施例】以下、図面を参照してこの発明の実施例を説明する。

【0026】図1にはこの発明の一実施例に係わるサーバ／クライアント形式の計算機システムが示されている。1a、1bはクライアントに対してファイルサービスを提供するファイルサーバ、2a、2b、2c、2dはファイルサーバ1a、1bのファイルを共同で利用するクライアント、3はファイルサーバとクライアントを

接続するネットワークである。4a～4fはファイルサーバ1aに存在するファイルシステム、4g～4lはファイルサーバ2bに存在するファイルシステムである。5はファイルサーバ1aに存在する仮想的ファイルシステムである。6a、6b、6c、6d、6eは、それぞれクライアント2a、2b、2c、2d、サーバ2bに存在するマウント表である。

【0027】ここで、仮想的ファイルシステム5は、システム内に存在する全ファイルシステム4a～4lを1つの木構造にまとめ上げ、仮想的に1つのファイルシステムとして見せる機能を持っている。よって、クライアント2a、2b、2c、2d、サーバ1bはシステム内にいくつのファイルシステムがどのサーバにあり、それらがどのように接続されているか知らないでよく、図1に示すように、ただファイルサーバ1a内の仮想的ファイルシステム5についての情報をそれぞれのマウント表に登録するだけでよい。

【0028】また、サーバやファイルシステムの数が変化しても仮想的ファイルシステム5の木構造が変わるもので、クライアント2a、2b、2c、2d、サーバ1bのマウント表はまったく変更する必要がなく、システム構成の変更が非常に容易である。次に、仮想的ファイルシステム5が、全ファイルシステム4a～4lを1つの木構造にまとめ上げたファイルシステムであることを説明する。図2は仮想的ファイルシステム5の内部構成である。

【0029】図2の仮想的ファイルシステム5において、7はシステム内の個々のファイル識別子であるVノードから仮想的iノード番号を得るV-i変換部、8は逆に仮想的iノード番号からそれに対応する実際のVノードを得るi-V変換部、9は各サーバの各ファイルシステム4a～4lが木構造のどの接続点で接続されているかを記憶するマウント表、10はバス名が与えられたとき、そのファイルが格納されている番号およびファイルシステム番号を求めるマウント表検索部、11はマウント表9に項目を登録、削除するためのマウント表管理部、12は仮想的ファイルシステム5に対するシステムコールをi-V変換部8、マウント表検索部10を用いて対応するサーバのファイルシステムに対するシステムコールに変換して実行させ、結果をV-i変換部7を用いて仮想的ファイルシステム5の結果に変換するシステムコール変換部である。

【0030】なお、V-i変換部7、i-V変換部8は図3に示す変換表を使って、それぞれ図5、図6に示すフローチャートに従って処理するものとする。また、マウント表検索部10は図4のフローチャート、システムコール変換部12は図7のフローチャートに従って処理するものとする。

【0031】ここで、クライアント2aから仮想的ファイルシステム5内のバス名が“/ABC/DIR1/F

ILE 2" であるファイルを読む場合を想定して、動作説明を行う。仮想的ファイルシステム5内のマウント表9は図2の通りで、図8に示すような大きな木構造を定義しているものとする。

【0032】まず、クライアント2aはファイル“/ABC/DIR1/FILE2”に対するオープンシステムコールを仮想的ファイルシステム5に要求する。これを仮想的ファイルシステム5のシステムコール変換部12が受け処理を行う。

【0033】システムコール変換部12は、第1に、マウント表検索部10を用いて、バス名が“/ABC/DIR1/FILE2”を格納するサーバ番号1a、ファイルシステム番号4bを求める。この検索処理は、図4のフローチャートに従って次のように行われる。

【0034】マウント表検索部10は、まず、検索するバス名（ここでは、“/ABC/DIR1/FILE2”）をセットし（ステップS11）、マウント表9に登録されている接続点と比較する（ステップS12）。バス名が途中まで一致するものが登録されていれば（ステップS13）、それを以前に一致した接続点と比較する（ステップS14）。今回一致した接続点の方が長い場合には（ステップS15）、その接続点に対応するマウント表9の情報（サーバ番号、ファイルシステム番号）を記憶する。ステップS12～S16の処理は、マウント表9の全エントリについて順番に実行される（ステップS17, S18）。

【0035】システムコール変換部12は、この様にして求められたサーバ番号1a、ファイルシステム番号4bで表わされるファイルシステムに対して、残っているバス名“/DIR1/FILE2”をオープンシステムコールとして発行する。サーバ番号1aのファイルシステム4b（Vノード）からは、“/DIR1/FILE2”で指定されるファイルに対応したiノード番号、例えば123が返却される。このiノード番号123は、ファイルシステム4b内でのローカルな番号である。

【0036】システムコール変換部12は、V-i変換部7を用いてVノード（サーバ番号1aのファイルシステム4b）からのiノード番号123を仮想的iノード番号623に変換し、それをクライアント2aに返す。ここで、V-i変換部7による変換処理は、図5のフローチャートに従って次のように行われる。

【0037】V-i変換部7は、まず、変換対象のVノードの情報をセットし（ステップS21）、Vノードのサーバ番号、ファイルシステム番号（ここでは、サーバ番号1a、ファイルシステム4b）を図3の変換表における各エントリのサーバ番号、ファイルシステム番号と比較する（ステップS22）。この比較は、Vノードのサーバ番号、ファイルシステム番号が一致するまで、エントリを更新しながら順次実行される（ステップS23, S24）。

【0038】Vノードのサーバ番号、ファイルシステム番号と一致するエントリが見つかると、そのエントリに登録されている仮想的iノード番号の開始値（ここでは、500）にVノードからのiノード番号（ここでは、123）が加えられ、これによって仮想的iノード番号（ $623 = 500 + 123$ ）が求められる（ステップS25）。

【0039】仮想的iノード番号623が通知されたクライアント2aは、その仮想的iノード番号623に対応するサーバシステムコールを仮想的ファイルシステム5に要求する。これもシステムコール変換部12が受け取り処理する。

【0040】システムコール変換部12は、第1に、i-V変換部8を用いて、仮想的iノード番号623から実際のVノード（ここでは、サーバ1a、ファイルシステム4b、iノード番号123）を求める。ここで、i-V変換部8による変換処理は、図6のフローチャートに従って次のように行われる。

【0041】i-V変換部8は、まず、仮想的iノード番号（ここでは、623）をセットし（ステップS31）、そのセットした仮想的iノード番号を図3の変換表における各エントリの仮想的iノード番号の範囲と比較する（ステップS32）。この比較は、セットした仮想的iノード番号（623）が属する範囲を持つ仮想的iノード番号範囲が見つかるまで、エントリを更新しながら順次実行される（ステップS33, S34）。

【0042】仮想的iノード番号（623）が属する仮想的iノード番号範囲（ここでは、500～1499）が見つかると、その仮想的iノード番号（623）から仮想的iノード番号範囲の開始番号（500）を引き、ローカルなiノード番号（ $123 = 623 - 500$ ）を求め、そしてそのエントリのサーバ番号（1a）、ファイルシステム番号（4b）を変換表から取り出す（ステップS35）。これによって、Vノード（サーバ1a、ファイルシステム番号4b、iノード番号123）が求められる。

【0043】このようにしてVノードが求められると、システムコール変換部12は、求まったサーバ1a、ファイルシステム4bに対してiノード番号123に対応するファイルのリードシステムコールを発行する。そして、最後に、システムコール変換部12は、サーバ1a、ファイルシステム4bから返って来るデータをそのままクライアント2aへ返す。図7のフローチャートには、以上説明したシステムコール変換部12による処理の手順が示されている。

【0044】すなわち、システムコール変換部12は、受け取った要求内にバス名が含まれているか否かを判断し（ステップS41）、バス名が含まれている場合は、マウント表検索部10を用いて、バス名をサーバ番号、ファイルシステム番号に変換する（ステップS42）。

2)。一方、バス名が含まれている場合には、システムコール変換部12は、受け取った要求内に仮想的iノード番号が含まれているか否かを判断し(ステップS4 3)、仮想的iノード番号が含まれている場合には、i-V変換部8を用いて、仮想的iノード番号をVノード(サーバ番号、ファイルシステム番号、iノード番号)に変換する(ステップS4 4)。

【0045】次いで、システムコール変換部12は、ステップS4 2またはステップS4 4で変換されたサーバ番号、ファイルシステム番号で、対応するリモートシステムコール(バス名が変換された場合はオープンシステムコール、仮想的iノード番号が変換された場合はリード/ライトシステムコール)を実行する(ステップS4 5)。

【0046】そして、システムコール変換部12は、システムコールの結果として返却された応答内にiノード番号が含まれているか否かを判断し(ステップS4 6)、iノード番号が含まれている場合には、そのiノード番号と、システムコールを実行したサーバ番号、ファイルシステム番号とを合わせてVノードとし、そのVノードをV-i変換部7を用いて仮想的iノード番号に変換して、それをバス名を含む要求を発行したクライアントに通知する(ステップS4 7)。

【0047】一方、システムコールの結果として返却された応答内にiノード番号が含まれていない場合には、その応答(例えば、リードデータ)に対して何も処理を加えずに、そのままその応答を仮想iノード番号を含む要求を発行したクライアントに通知する。

【0048】以上のように、この実施例においては、クライアント2aからは仮想的ファイルシステム5に対するファイルオープンやファイルリードを1つの大きな木構造に対して行っているように見え、実際には複数のファイルシステムから構成されていることに気付かない。よって、仮想的ファイルシステム5はシステム内の全ファイルシステムを1つの木構造にまとめ上げたものとしてクライアントから見える。

【0049】よって、先に説明したように、各クライアント2a、2b、2c、2d、サーバ1bは、ファイルサーバ1a内の仮想的ファイルシステム5だけをそれぞれのマウント表に登録するだけで全ファイルへのアクセスが可能となる。また、サーバやファイルシステムの数が変化してもクライアント2a、2b、2c、2d、サーバ1bは何ら変更する必要がない。

【0050】また、サーバやファイルシステムの変更は仮想的ファイルシステム5内のマウント表9だけをマウント表管理部11を用いて変更するだけでよく、その変更も非常に簡単である。なお、図3の変換表は各ファイルシステムの最大iノード番号から容易に作成でき、最大iノード番号も簡単に求められるので、自動的作成が可能である。

【0051】なお、仮想的ファイルシステムのコピーを他のサーバにも置いたり、マウント表の一部をクライアント側にキャッシングして仮想的ファイルシステムをクライアントで実現することも可能である。

【0052】また、以下の説明では、1つの仮想的iノードに対して1つのVノードを対応させていたが、図9に示すように、1つの仮想的iノード番号を複数のVノードに対応させることも可能である。

【0053】この図9では、仮想的iノード番号(1)を、4個のVノード1~Vノード4に対応付けた場合が示されている。ここでは、Vノード1として図1のサーバ1a、ファイルシステム4a、Vノード2としてサーバ1a、ファイルシステム4b、Vノード3としてサーバ1a、ファイルシステム4c、Vノード4としてサーバ1a、ファイルシステム4dが割り当てられている。

【0054】この場合には、1つの仮想的iノード番号(1)に対応する4個のVノードのファイル全部を集めたものが、その仮想的iノードで表わされる1つのファイルを構成しているものと考えられる。このように複数のファイルを1つのファイルとして構成するための手法としては、次のようなものが考えられる。

【0055】(ア) 図10に示すように、Vノード1からVノード4までのファイルを直列に接続する。この場合、ファイル作成時には、Vノード1のファイルシステムがいっぱいになるまで書いて、次にVノード2のファイルシステムがいっぱいになるまで書く。これをVノード4まで続ける。

【0056】(イ) 図11に示すように、1つの仮想的iノード番号(1)に対応するファイルを、Vノード1からVノード4までの複数のファイルシステムにあるストライプ間隔を置いてブロック単位で分散して割り当てる(ここでは、1つの仮想的iノード番号のファイルが、Vノード1のファイルの第1ブロックB1、Vノード2のファイルの第2ブロックB2、Vノード3のファイルの第3ブロックB3、Vノード4のファイルの第4ブロックB4、Vノード1のファイルの第5ブロックB5、Vノード2のファイルの第6ブロックB6、Vノード3のファイルの第7ブロックB7、Vノード4のファイルの第8ブロックB8、Vノード1のファイルの第9ブロックB9、およびVノード2のファイルの第10ブロックB10に対応している)。などが考えられる。

【0057】これらの対応関係は、仮想的iノードに対するシステムコールをどのように複数のVノードに対するシステムコールに変換するかによって決まるものである。しかし、どちらの場合でも、1つのファイルを実際には複数のファイルに分割して、複数のファイルシステムに格納できる。よって、負荷分散と、効率的ファイル配置を容易に実現できる。

【0058】

【発明の効果】以上のように、この発明によれば、ファ

イルの実際の配置状態に関係なくクライアントは1つの仮想ファイルシステムだけをアクセスする事によって任意のファイルアクセスを実現できるようになりし、クライアントにおけるマウント表の管理を不要とすることによって、ファイルシステムやサーバの追加等の変更も仮想的ファイルシステム内のマウント表を変更するだけでよく、各クライアントはまったく変更する必要がない。さらに、ファイルを仮想的にアクセスできるので、複数のファイルから1つのファイルを構成してクライアントに見せることができ、負荷分散や効果的ファイル配置が可能となる。

## 【図面の簡単な説明】

【図1】この発明の一実施例に係るサーバ/クライアント形式の計算機システムの構成を示すブロック図。

【図2】同実施例の計算機システムに設けられた仮想ファイルシステムの構成を示すブロック図。

【図3】図2の仮想ファイルシステムに設けられるV-i, i-V変換用の変換表を示す図。

【図4】図2の仮想ファイルシステムにおけるマウント表検索処理の手順を示すフローチャート。

【図5】図2の仮想ファイルシステムにおけるV-i変換処理の手順を示すフローチャート。

【図6】図2の仮想ファイルシステムにおけるi-V変換処理の手順を示すフローチャート。

【図7】図2の仮想ファイルシステムにおけるシステムコール変換処理の手順を示すフローチャート。 \*

\* 【図8】図2の仮想ファイルシステムで管理されるファイルの木構造を示す図。

【図9】同実施例における仮想的iノードと複数のVノードとの対応関係を示す図。

【図10】同実施例において複数ファイルから1つのファイルを構成する場合の一例を示す図。

【図11】同実施例において複数ファイルから1つのファイルを構成する場合の他の例を示す図。

【図12】従来のサーバ/クライアント形式の計算機システムの構成を示すブロック図。

【図13】従来の計算機システムにおけるファイルの木構造を示す図。

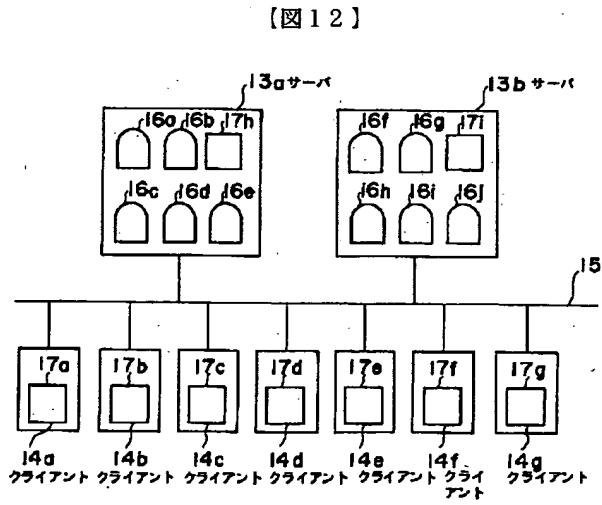
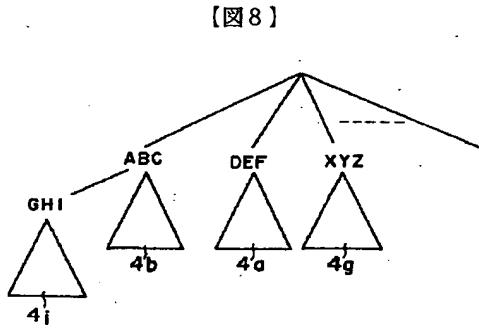
【図14】従来の計算機システムにおいて各サーバおよび各クライアントに設けられるマウント表を示す図。

【図15】従来の計算機システムにおけるファイルシステムが持つ部分木構造の一例を示す図。

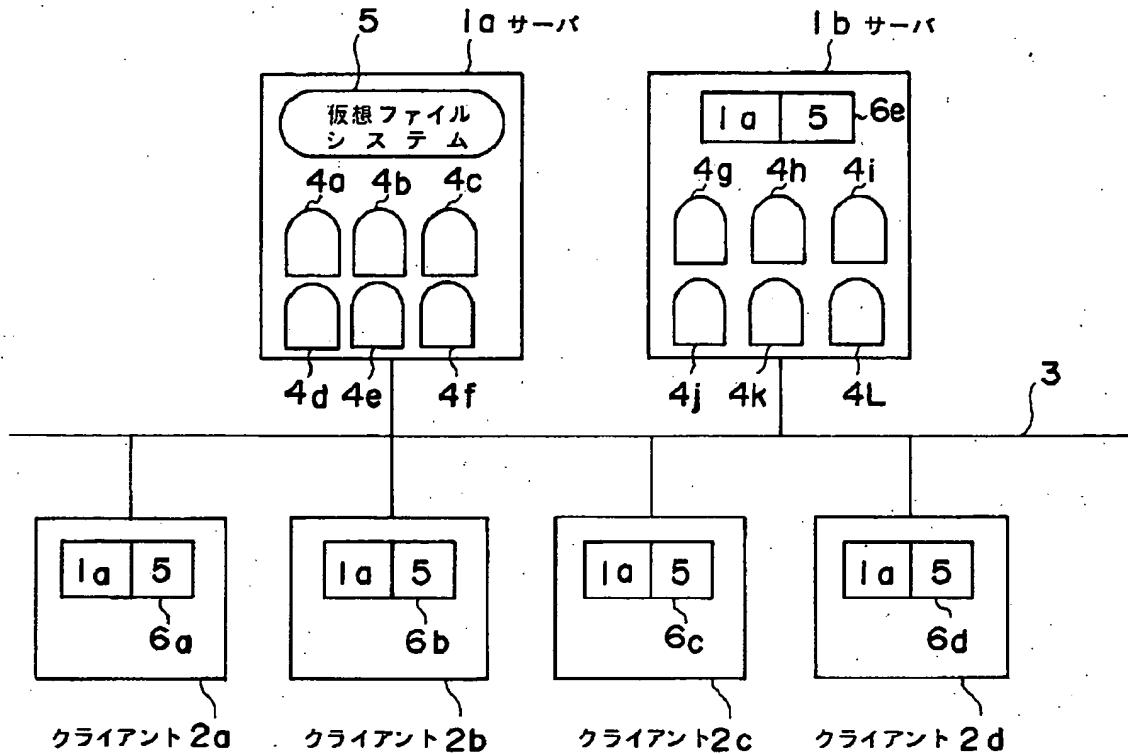
【図16】従来の計算機システムにおいて各クライアントが管理するVノードを示す図。

## 【符号の説明】

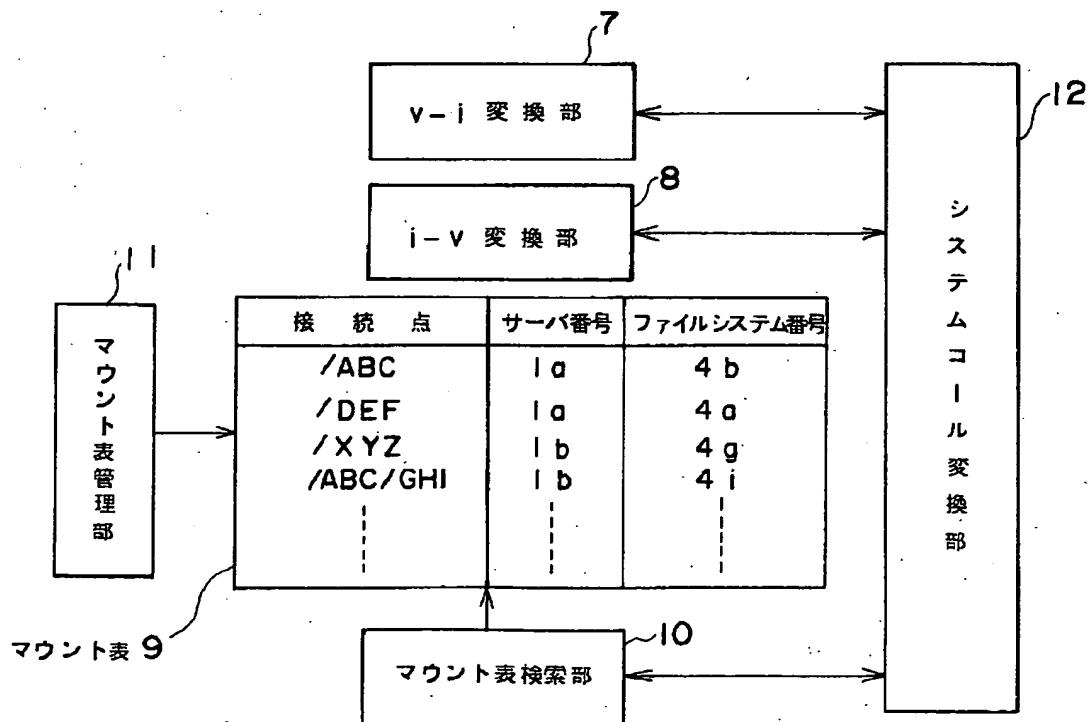
- 20 1a, 1b…ファイルサーバ、2a, 2b, 2c, 2d…クライアント、3…ネットワーク、4a～4L…ファイルシステム、5…仮想的ファイルシステム、6a, 6b～6e…マウント表、7…V-i変換部、8…i-V変換部、9…マウント表、10…マウント表検索部、11…マウント表管理部、12…システムコール変換部。



【図1】



【図2】



【図3】

| サーバ番号 | ファイルシステム番号 | 最大ノード番号 | 仮想的ノード番号  |
|-------|------------|---------|-----------|
| 1a    | 4a         | 500     | 0~499     |
| 1a    | 4b         | 1000    | 500~1499  |
| 1a    | 4c         | 300     | 1500~1799 |
| 1a    | 4d         | 400     | 1800~2199 |
| 1a    | 4e         | 800     | 2200~2999 |
| 1a    | 4f         | 1100    | 3000~4099 |
| 1b    | 4g         | 2000    | 4100~6099 |
| 1b    | 4h         | 1500    | 6100~7599 |
| ⋮     | ⋮          | ⋮       | ⋮         |

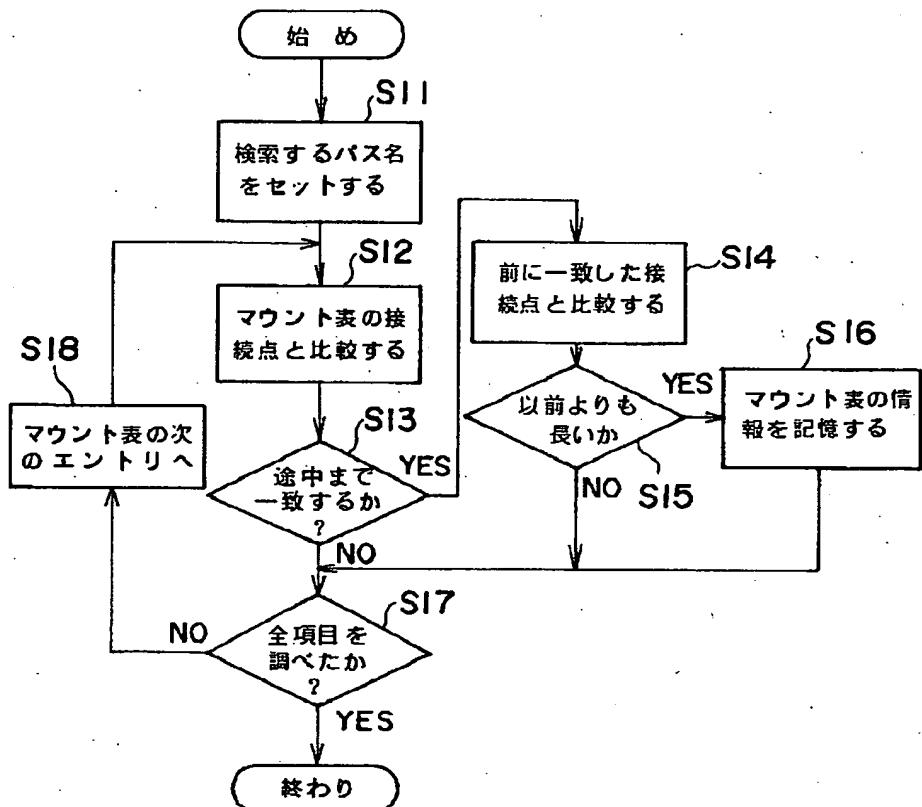
【図9】

| 仮想的ノード | vノード1  | vノード2  | vノード3  | vノード4  |
|--------|--------|--------|--------|--------|
| 1      | 1a, 4a | 1a, 4b | 1a, 4c | 1a, 4d |
| ⋮      | ⋮      | ⋮      | ⋮      | ⋮      |
| ⋮      | ⋮      | ⋮      | ⋮      | ⋮      |
| ⋮      | ⋮      | ⋮      | ⋮      | ⋮      |
| ⋮      | ⋮      | ⋮      | ⋮      | ⋮      |
| ⋮      | ⋮      | ⋮      | ⋮      | ⋮      |

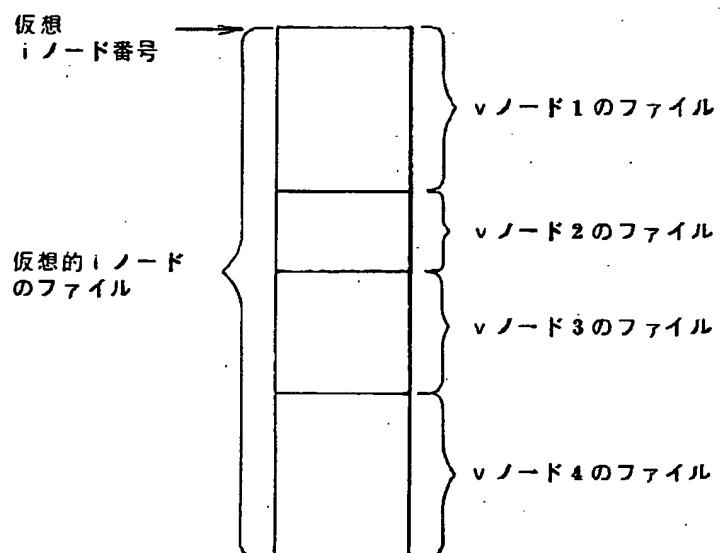
【図16】

| サーバ番号 | ファイルシステム番号 | ノード番号 |
|-------|------------|-------|
| 13a   | 16a        | 123   |

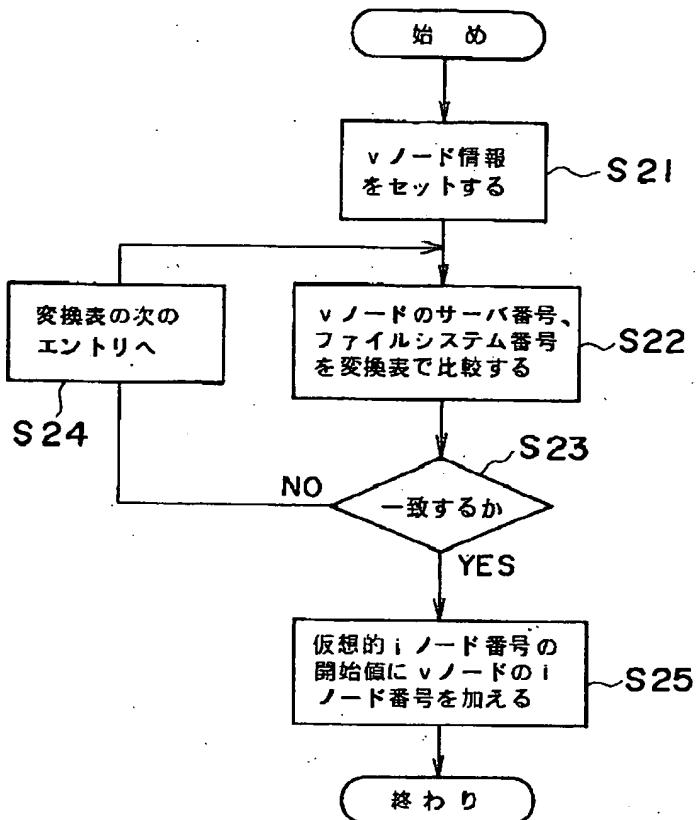
【図4】



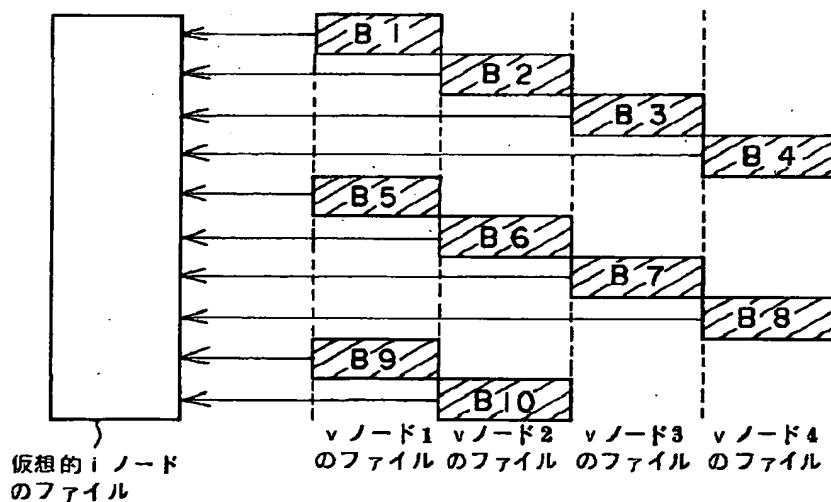
【図10】



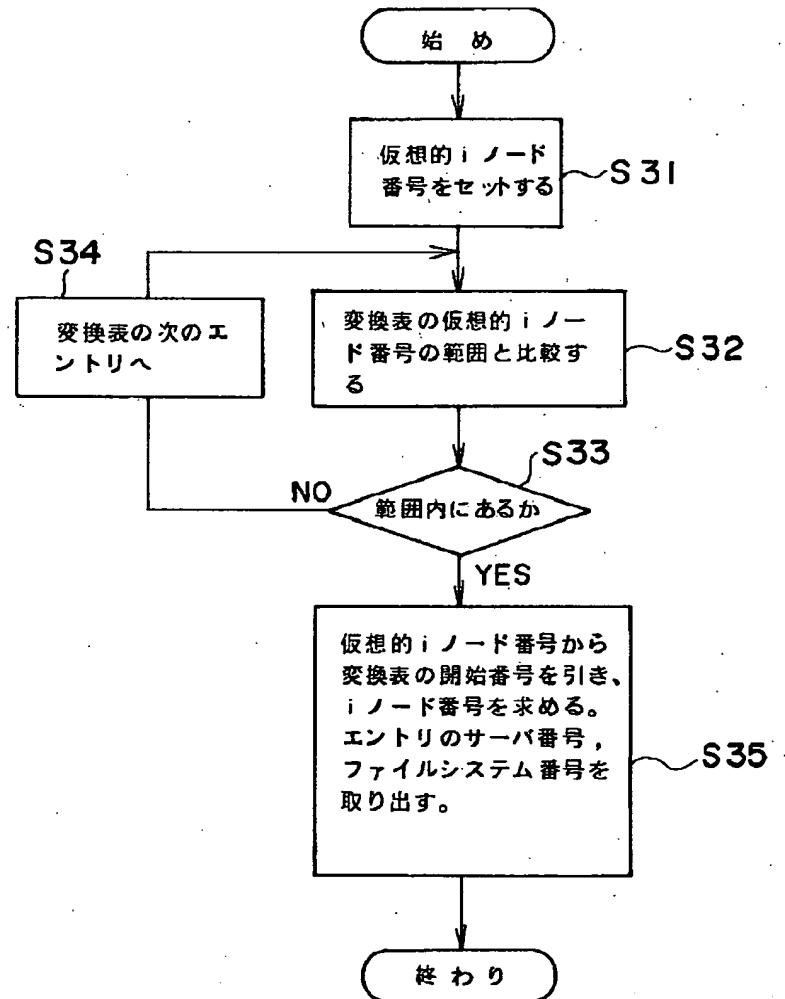
【図5】



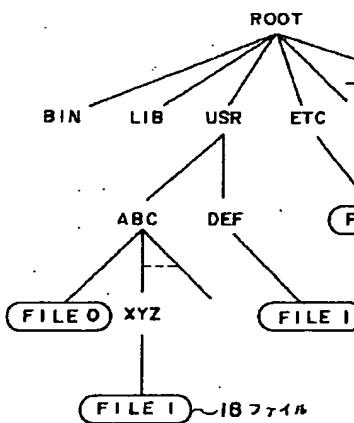
【図11】



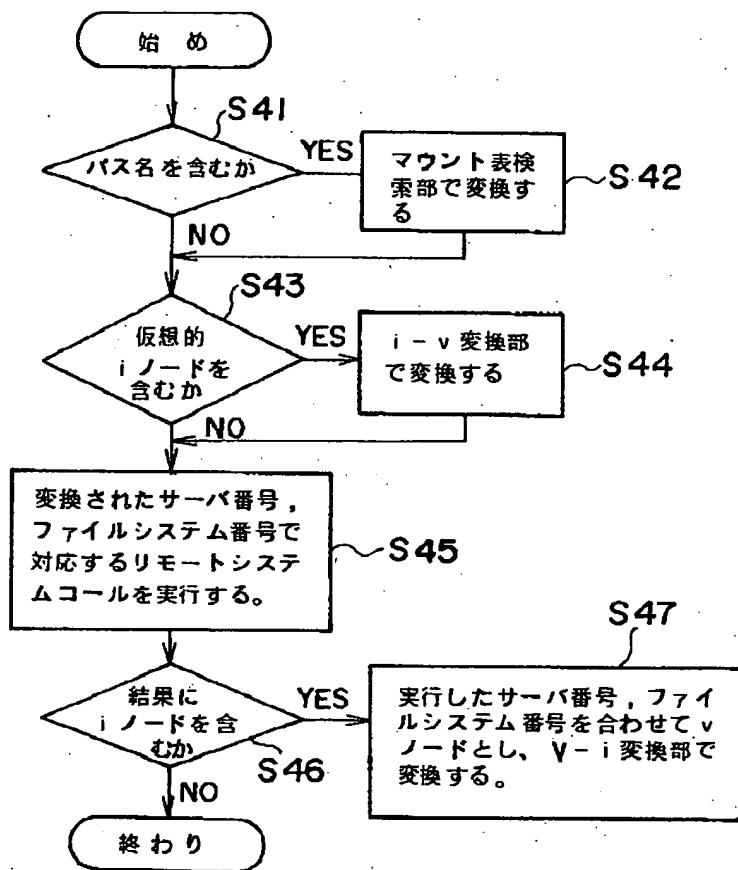
【図6】



【図13】



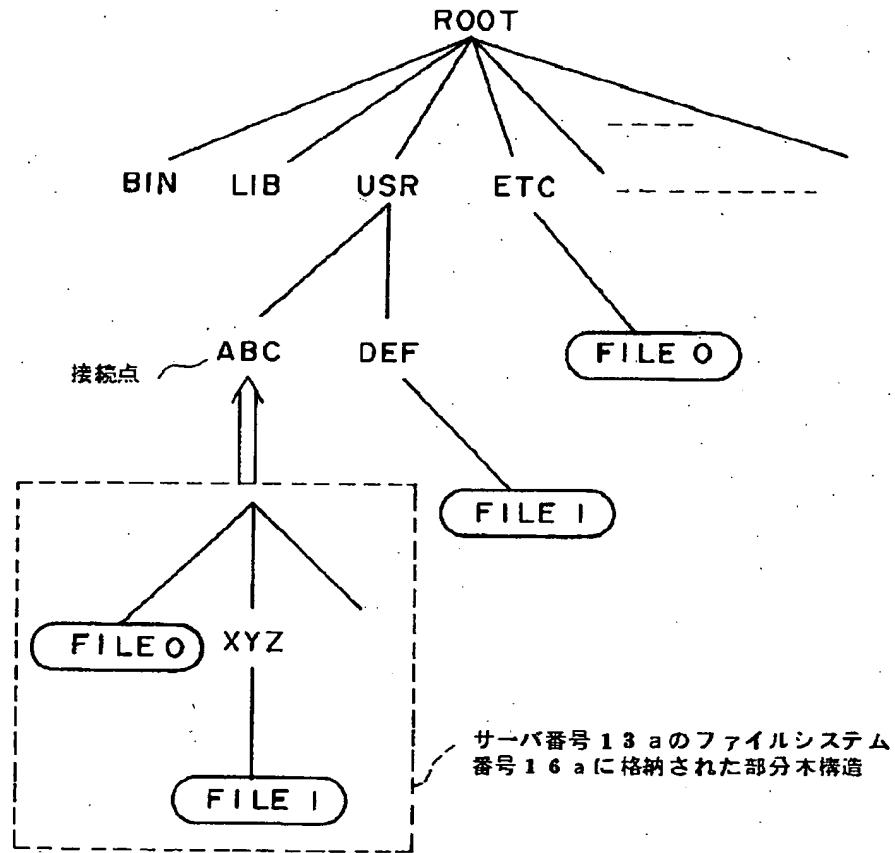
【図7】



【図14】

|        | 接続点      | サーバ番号 | ファイルシステム番号 |
|--------|----------|-------|------------|
| エントリ 1 | /USR/ABC | 13a   | 16a        |
|        | /BIN     | 13a   | 16d        |
|        | /LIB     | 13b   | 16h        |

【図15】



【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成11年(1999)8月27日

【公開番号】特開平5-241934

【公開日】平成5年(1993)9月21日

【年通号数】公開特許公報5-2420

【出願番号】特願平4-42508

【国際特許分類第6版】

|      |       |     |
|------|-------|-----|
| G06F | 12/00 | 545 |
|      | 15/16 | 370 |

【F I】

|      |       |       |
|------|-------|-------|
| G06F | 12/00 | 545 B |
|      | 15/16 | 370 M |

#### 【手続補正書】

【提出日】平成10年9月16日

#### 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】発明の名称

【補正方法】変更

#### 【補正内容】

【発明の名称】計算機システムおよびファイル管理办法

#### 【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

#### 【補正内容】

#### 【特許請求の範囲】

【請求項1】1以上のサーバ計算機とこのサーバ計算機が提供するファイルサービスを共有する1以上のクライアント計算機から構成されたサーバ/クライアント型の計算機システムにおいて、

前記サーバ計算機の少なくとも1つは、前記各クライアントからのファイルアクセス要求をすべて受け付けるファイル管理手段を具備し、

このファイル管理手段は、前記計算機システムの全てのファイルを1つの木構造で管理する手段と、前記各クライアントからアクセス要求されたファイルが存在するサーバ計算機のファイルシステムを前記木構造に従って調べ、そのサーバ計算機のファイルシステムにアクセス要求を発行する手段と、前記アクセス要求されたファイルが存在するサーバ計算機のファイルシステムから返却されるデータを前記アクセス要求元の各クライアントに送信する手段とを具備することを特徴とするサーバ/クライアント型の計算機システム。

【請求項2】木構造でファイル管理を行うファイルサーバを含む1以上のサーバ計算機と、前記ファイルサーバが提供するファイルサービスを共有する1以上のク

イアント計算機から構成され、個々のファイルをサーバ番号、サーバ内のファイルシステム番号、そのファイルシステム内で固有のファイル識別番号から構成される識別子を用いてアクセスする計算機システムにおいて、前記木構造の接続点を記憶するマウント表と、このマウント表に対する接続点の登録・削除を行うマウント表管理手段と、前記マウント表を調べ、接続先を求めるマウント検索手段と、前記識別子から仮想的なファイル識別番号へ変換する第1の変換手段と、前記仮想的なファイル識別番号から前記識別子へ変換する第2の変換手段と、前記仮想的なファイル識別番号に対するシステムコールを対応する前記識別子に対するシステムコールへ変換するシステムコール変換手段から構成される仮想的ファイルシステムを前記サーバの少なくとも1つに具備せしめ、

前記サーバおよびクライアントは各々のマウント表に前記仮想的ファイルシステムへの接続点を登録することを特徴とするサーバ/クライアント型の計算機システム。

【請求項3】前記第2の変換手段は、前記各仮想的なファイル識別番号に対して複数の前記識別子を割り当てる手段を具備し、前記システムコール変換手段は、前記各仮想的なファイル識別番号に対するシステムコールを前記複数の識別子に対するシステムコールに変換する手段を具備することを特徴とする請求項2記載の計算機システム。

【請求項4】1以上のサーバ計算機とこのサーバ計算機が提供するファイルサービスを共有する1以上のクライアント計算機から構成されたサーバ/クライアント型の計算機システムにおいて、前記サーバ計算機の少なくとも1つに設けられたファイル管理手段を用いて前記計算機システム上のファイルを管理するファイル管理办法であって、

前記計算機システムの全てのファイルを1つの木構造で管理するステップと、

前記各クライアントからアクセス要求されたファイルが存在するサーバ計算機のファイルシステムを前記木構造に従って調べ、そのサーバ計算機のファイルシステムにアクセス要求を発行するステップと、

前記アクセス要求されたファイルが存在するサーバ計算機のファイルシステムから返却されるデータを前記アクセス要求元の各クライアントに送信するステップとを具備することを特徴とするファイル管理方法。

【請求項5】 木構造でファイル管理を行うファイルサーバを含む1以上のサーバ計算機と、前記ファイルサーバが提供するファイルサービスを共有する1以上のクライアント計算機から構成され、個々のファイルをサーバ番号、サーバ内のファイルシステム番号、そのファイルシステム内で固有のファイル識別番号から構成される識別子を用いてアクセスする計算機システムにおいて、前記サーバ計算機に設けられた仮想的ファイルシステムを用いて前記計算機システム上のファイルを管理するファイル管理方法であって、

前記木構造の接続点を記憶するマウント表に対して接続点の登録・削除を行うステップと、

前記マウント表を調べ、接続先を求めるマウント検索ステップと、

前記識別子から仮想的なファイル識別番号へ変換する第1の変換ステップと、

前記仮想的なファイル識別番号から前記識別子へ変換する第2の変換ステップと、

前記仮想的なファイル識別番号に対するシステムコールを対応する前記識別子に対するシステムコールへ変換す

るシステムコール変換ステップとを具備し、

前記サーバおよびクライアントは各々のマウント表に前記仮想的ファイルシステムへの接続点を登録することを特徴とするファイル管理方法。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0001

【補正方法】変更

【補正内容】

【0001】

【産業上の利用分野】この発明はサーバ／クライアント形式でファイルを共有する計算機システムおよび同計算機システムに適用されるファイル管理方法に関する。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0019

【補正方法】変更

【補正内容】

【0019】この発明はこのような点に鑑みてなされたもので、ファイルの実際の配置状態に関係なくクライアントは1つの仮想ファイルシステムだけをアクセスする事によって任意のファイルアクセスを実現できるようにし、クライアントにおけるマウント表の管理を不要とすることによって、サーバの追加等によるシステム変更やファイルの移動を伴う柔軟なファイル管理を行うことができる計算機システムおよびファイル管理方法を提供することを目的とする。